Release Planning Document

Arana Charlebois, Tristan
Carvalho, Cam
Corbett, Cole
Kassie, Turner
Pauls, Andrew
Wallace, Jordan

Cosc 4P02
Dr. Naser Ezzati-Jivan

January 19, 2025

Contents

1 Introduction

The system that we are aiming to provide is one that can determine a route for travel, based on the coordinates or way-points on a map, Essentially, we aim to deliver an acceptable answer to the known traveling salesman problem. In addition, we want to allow users to adjust routes based on factors such as the number of travelers and minimizing the route's completion time. This type of system is particularly valuable for logistics companies that manage multiple drivers and delivery routes, but it is also valuable for users who make multiple stops and are interested in an efficient route.

The main goals that we share as a team is to produce a service that has a practical use, (optimizing routes for logistics), but also a system that is usable, reliable, and available. This document will outline the user stories, the requirements for the system, show the product backlog, outline tasks and their corresponding user stories and explain any challenges that we may face. In addition, we will outline the plans for the initial sprints.

2 Functional Requirements of the System

- A user can define a list of addresses, or coordinates on the map for the system to determine the optimal route.
- A user can register as an organization and have the ability to set routes for users who are part of their organization.
- Users can save routes for later use after the route has been computed.
- A user can define whether they would like to prioritize a certain metric, in particular, the quickest or most fuel efficient route.
- A user can define the mode of transportation, such as a vehicle, a semi-truck, biking, walking...
- A user can specify route constraints like perhaps avoiding highways and tolls.
- Organizations can add users and assign different roles to them such as driver or admin.

3 Non-Functional Requirements of the System

- The system should be intuitive for users without prior experience.
- The system will efficiently determine a route based on inputs, quickly returning a valid route.

- The system allows for numerous users to be able to handle multiple route requests at once.
- The system should return the same or similar results for a given input.
- The system will use a genetic algorithm to determine "good" routes for the user and find a good balance of accuracy and speed.
- When the system fails to find a route, this information should be communicated to the user.

4 User Stories

User story 1 (Metric Optimization): As a user, I would like to adjust my route to prioritize fuel efficiency or reduced completion time.
Acceptance Criteria: A user can adjust their results for fuel optimization, or completing the cycle in the minimal amount of time.
Priority: High
Estimation: 12 hours

- Be able to retrieve fuel data and calculate travel time based on the route selected.
- Develop GA for determining a valid route of travel, based on factors other than the distance between the locations.
- Add a section to the route options that allows the user to pick the preferred route adjustment.
- Recalculate and display a new route based on user metrics.

User story 2 (Single TSP GA): As an individual user, I want to enter stops along a route and receive a route optimized for distance, with constraints on locations and roads.
Acceptance Criteria: GA can determine a viable route for travel, in an efficient amount of time.
Priority: High
Estimation: 17 hours

- Implement GA for single-TSP optimization (core logic).
- Design input/output interfaces (API endpoints).
- Integrate constraint validation (e.g., invalid locations).
- Write tests for GA logic and API endpoints.

User story 3 (Multi TSP GA): As an enterprise user, I want to enter stops and a number of drivers, and have a system of routes generated that are optimized for concurrent travel time.
Acceptance Criteria: Viable routes based on multiple TSP, and returned efficiently.
Priority: Medium
Estimation: 21 hours

- Extend GA to handle multi-TSP optimization.
- Implement a workload distribution (e.g., dividing tasks by vehicle).
- Design API endpoints for multi-TSP input/output.
- Write tests for multi-TSP logic and API.

User story 4 (Account Creation): As a user, I want to be able to create an account or use a provider account.
Acceptance Criteria: The user can create, login, and logout to unique account.
Priority: High
Estimation: 7 hours

- Make database for account storage.
- Create login/logout page.
- Verify login information.

User story 5 (Account Management): As a user, I want to be able to manage my account (add recovery email, reset password, etc.).
Acceptance Criteria: User can edit the details of the account, reset account, and login with new credentials.
Priority: High
Estimation: 5 hours

- Users can add a recovery email to account.
- Users can choose to enable 2FA on account.
- Users can choose to reset password to email.

User story 6 (Route Distance): As a user, I would like to search for locations by name, rather than placing waypoints on the map.
Acceptance Criteria: Search for locations can autocomplete, and when a user selects it, it adds it to the list of stops.
Priority: High
Estimation: 10 hours

- Create a search bar (with auto-complete) to allow users to search for locations.
- Create an option for the user to either use the current location or "from" location.
- Display route distance to the user.

User story 7 (Route Assignment): As a delivery supervisor, I want to be able to assign different nodes to each traveler and have the ability to see their routes so that I can efficiently distribute the workload.
Acceptance Criteria: Calculate and display multiple routes concurrently, each one distinguishable by color.
Priority: Medium
Estimation: 6 hours

- Create an interface for the user to change the number of vehicles.
- Calculate and display multiple routes concurrently, each one distinguishable by color.

User story 8 (Adding to Route): As a mother, I want to add multiple stops (nodes) to my family road trip so that I don't experience too much fatigue while driving.
Acceptance Criteria: One should be capable of selecting an existing route, and adding stops, with the new stops being included in a newly generated route.
Priority: Medium
Estimation: 8 hours

- Change UI to allow the user to edit the route.
- Create an interface that enables the user to add and remove nodes to/from the path.
- Display recalculated route based on nodes added by the user.

User story 9 (Route Visualization): As a user, I would like to have a visualized route, rather than a list of stops.
Acceptance Criteria: User's route can be seen, similar to the output of directions on Google Maps.
Priority: Medium
Estimation: 10 hours

- Use mapping API (such as Google Maps) to be able to display the route to the user.

User story 10 (Ending Location): As a user, I'd like to mark a location as an ending location to ensure the route planning ends at a designated location.
Acceptance Criteria: Routes are generated in the same manner, but rather than returning to the original starting location, the route ends at the end location, specified by the user.
Priority: Low
Estimation: 10 hours

- Add support for specifying an ending location in the route input payload.
- Update the optimization algorithm to handle the ending location as fixed.
- Ensure the input endpoint validates that the ending location exists within the provided list of locations.
- Write unit tests for the updated API and logic.

User story 11 (Real-time updates): As a truck driver, I want to receive real-time updates on my route to inform my supervisor about any possible delays.
Acceptance Criteria: Recalculate route automatically based on real-time traffic updates.
Priority: Medium
Estimation: 30 hours

- Display any route changes including traffic and road closures to the user.
- Add a pop-up with an alert sound to notify the user about route time updates.

User story 12 (Storing Routes): As a user, I want to be able to save the routes I am planning.
Acceptance Criteria: User can load previously stored routes from database to front-end.
Priority: High
Estimation: 15 hours

- User can select to save calculated route in front-end.
- Route is sent and stored in standard format in database.
- User can choose to load saved routes from database.
- Database can send properly formatted data to front-end.
- Route properly displays on front-end mapping software.

User story 13 (Batch Processing): As an enterprise, I'd like to preemptively submit a group of jobs so routes can be readily dispatched to drivers when they are needed.
Acceptance Criteria: Enterprises are able to submit numerous sets of stops and receive the optimal route for each set.
Priority: Low
Estimation: 20 hours

- Implement batch processing logic (e.g., bulk job queuing).
- Design API for batch submission and status monitoring.
- Optimize batch processing to handle large datasets efficiently.
- Write tests for batch submission and processing.

User story 14 (Analytics): As a server administrator, I'd like to view usage statistics and history to indicate server usage, and guide future resource allocation.
Priority: Low
Estimation: 15 hours

- Design data model for logging user activity and usage stats.
- Implement backend logic for logging and storing analytics data.
- Create APIs or dashboards for viewing analytics.
- Write tests for data logging and visualization.

User story 15 (Route Constraints): As a user, I want to be able to customize routes to have constraints, such as vehicle clearance for tunnels, or avoiding bridges and borders.
Acceptance Criteria: Users can incorporate custom constraints into their route.
Priority: Medium
Estimation: 20 hours

- Extend GA to incorporate constraints (e.g., avoiding specific roads).
- Update API to accept constraint parameters.
- Implement validation for custom constraints.
- Write tests for constraint handling.

User story 16 (Exporting Routes): As a user, I want to be able to export my routes in a standard format for use in other apps.
Acceptance Criteria: Users can choose to export routes to other related apps (OSM, Mapbox, etc.).
Priority: Low
Estimation: 10 hours

- Integration of other popular apps to Reroute.

5 Sprint Outlines

We will outline the user stories, their associated tasks, and any additional tasks within each sprint, noting that the plan is tentative. The corresponding Trello board can be found in this section, after outlining each sprint.

Sprint 1

- User Story 1 (Metric Optimization)
- User Story 2 (Single TSP GA)
- User Story 4 (Account Creation)
- User Story 5 (Account Management)
- User Story 6 (Route Distance)
- Task: Integrate RabbitMQ for Route Calculation Tasks
- Task: Containerize the Application with Docker
- Task: API Planning

Sprint 2

- User Story 3 (Multi TSP GA)
- User Story 7 (Route Assignment)
- User Story 8 (Adding to Route)
- User Story 9 (Route Visualization)
- Task: Deploy and Scale with Kubernetes

Sprint 3

- User Story 10 (Ending Location)
- User Story 11 (Real-time updates)
- User Story 12 (Storing Routes)
- Task: Integrate Stripe for Billing and Subscription Management

Sprint 4

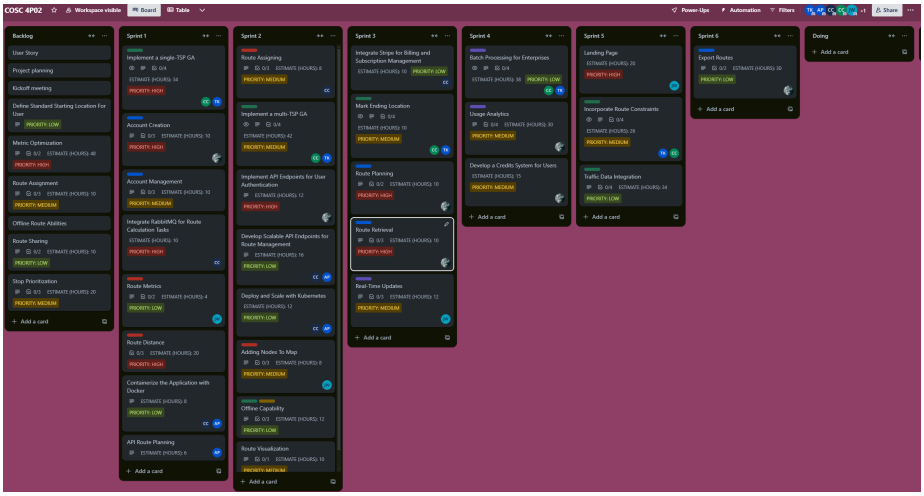
- User Story 13 (Batch Processing)
- User Story 14 (Analytics)
- Task: Develop a Credits System for Users

Sprint 5

- User Story 15 (Route Constraints)
- Task: General testing (Stress testing, Volume testing, Capacity testing...)
- Task: Enhance landing page

Sprint 6

- User Story 16 (Exporting Routes)
- Task: Final Report



For readers with a Trello account, we invite you to view the Trello board here.

6 Challenges and Planning

The main issue that we are facing with this project is that our final system is quite complex and has many requirements. Development will likely take longer than planned, and balancing a project of this scale will be a challenge for the team. Despite this, we know that we are determined and ready to work. Our current plan is to refine our requirements and adjust the workload for each sprint, since, as shown by the outline, there is a lot of work for the early sprints and less in the latter half. Also, we plan to start development on the GA, user database, and landing page for the system.

7 Release Planning Meeting Minutes

- Shared user stories that each team member produced

- Spoke with TA about current plan for the system
- Displayed current work developed by Cam to TA
- Created Trello board, and all team members entered user stories
- Trello board refined by Andrew to show tenative release plans
- Started discussion and planning for the GA (shown below)